ELSEVIER

Journal of Computational and Applied Mathematics 79 (1997) 87–99

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Four step methods for $y''=f(x,y)$

L.Gr. Ixaru*, M. Rizea

Institute of Physics and Nuclear Engineering, Department of Theoretical Physics and Computer Center, PO Box MG-6, Măgurele, Bucharest, R-76900, Romania

Received 7 October 1995; revised 10 November 1996

Abstract

A systematic investigation is undertaken here on the possible practical consequences of the fact that the linear four step methods for $y''=f(x,y)$ form a family with a certain structure. The mentioned property is found to be surprisingly rich in consequences, and three of them are of particular importance: (i) drastic reduction in the number of iterations at each step, (ii) increased flexibility with respect to the start of integration and to the modification of the step size during the integration process, (iii) possibility of direct and simple appraisal of the local truncation error.

Keywords: Second order differential equations; Multi-step methods; Consistency; Stability; Predictor–corrector mode; Error control

AMS classification: 65L05, 65L06, 65L20, 65L50

1. Introduction

Significant efforts were undertaken over the years to promote the linear four step methods as highly competitive solvers for systems of the form $y''=f(x,y)$.

The Schrödinger equation received particular attention in this context, see [3, 8, 11–13], and for this equation the four step methods are particularly attractive. Their algorithms are simple enough to allow completing short, unsophisticated programs without difficulty and, moreover, the accuracy of the results is better than for similar programs based on other simple algorithms. For example, the order of the four step methods is six while for popular methods like Numerov, de Vogelaere or the classical Runge–Kutta method, the order is four, see [7]. The same is true for the more recent method proposed by Allison and Huang [1]. We also mention that various ways to improve

* Corresponding author. E-mail: ixaru@roifa.ifa.ro.

PII S0377-0427(96)00165-3

further the accuracy of the four step methods for the Schrödinger equation were considered in a series of recent papers [14–18]. Several versions of possible fittings were exploited, among them the exponential, Bessel and phase fittings.

In spite of such a good behaviour with respect to the Schrödinger equation, many people think that the four step methods are too rigid to remain competitive in the general case $y''=f(x,y)$; in particular, to form a basis for writing more demanding codes. The autonomous calculation of the starting values, the error control and the modification of the stepsize during the integration process, for instance, are seen as requirements which cannot be conveniently answered by these methods.

The purpose of this paper consists in showing that such fears are unjustified. We actually present a procedure to autonomously generate the additional starting values when the initial data are known only at two points, a procedure to halve the stepsize when necessary and also a suited predictor–corrector pair which enables a direct evaluation of the local error. With these improvements, the four step methods become really competitive with the existing methods of comparable order, as it is illustrated on two rather severe test cases in which the Runge–Kutta based method DOPRIN (this is a 6(7)th order one step method) [5] is taken for reference.

All these improvements became possible upon recognizing that the four step methods form a family. As a matter of fact, this property is known for some time, see for instance [10], but it was never investigated with practical consequences in mind. What we do in this paper represents therefore the first investigation along this line and the new procedures enumerated above should be seen as those which seem to us the most important ones from the point of view of practical consequences.

2. The family of four step methods

As announced, in this paper we focus on the numerical solution of the initial value problem

$$y''=f(x,y), \quad x\in[x_{\min},x_{\max}], \quad y(x_{\min})=y_0, \quad y'(x_{\min})=y'_0 \tag{2.1}$$

by four step methods.

A four step method is a method with the algorithm

$$\begin{aligned} &a_0y_{n+2}+a_1y_{n+1}+a_2y_n+a_3y_{n-1}+a_4y_{n-2}\\ &\quad=h^2[b_0f_{n+2}+b_1f_{n+1}+b_2f_n+b_3f_{n-1}+b_4f_{n-2}], \end{aligned} \tag{2.2}$$

where

$$f_i=f(x_i,y_i).$$

The values of the ten weights a_i and b_i are derived by imposing the two necessary and sufficient conditions for convergence, i.e. consistency and stability, see [6, 7] for the general theory. Since Eq. (2.2) is homogeneous in the weights, one weight can be fixed from the very beginning without any loss in generality; we put $a_0=1$. The other nine weights remain open for further determination in terms of the same number of conditions. One condition is simply the stability condition while the other eight collectively account for the consistency condition.

2.1. Consistency condition

We associate to Eq. (2.2) the functional

$$F[z(x);h] = z(x+2h) + a_1 z(x+h) + a_2 z(x) + a_3 z(x-h) + a_4 z(x-2h)$$
$$-h^2[b_0 z''(x+2h) + b_1 z''(x+h) + b_2 z''(x) + b_3 z''(x-h) + b_4 z''(x-2h)] \tag{2.3}$$

and require that F identically vanishes for all x and h when $z(x)$ is any of the following eight functions:

$$1, x, x^2, x^3, x^4, x^5, x^6, x^7. \tag{2.4}$$

A linear system of eight equations for nine unknowns results (an outline of the procedure is given in [7, p. 173]) which admits a unique solution if one unknown is assumed to be given. We take $a_2 = -2q$ where q is a free parameter and thus we obtain the following set of weights:

$$a_0 = a_4 = 1, \qquad a_1 = a_3 = q - 1, \qquad a_2 = -2q,$$
$$b_0 = b_4 = (17 - q)/240, \qquad b_1 = b_3 = (29 + 3q)/30, \qquad b_2 = (111 + 97q)/120. \tag{2.5}$$

This tells us that the four step schemes form a one-parameter family. Each scheme in this family is identified by the value assigned to q and in the following it will be abbreviated simply as $S(q)$.

If $z(x)$ is a function which admits Taylor series representation we have

$$F[z(x);h] = h^8 C(q) z^{(8)}(x) + O(h^{10})$$

with

$$C(q) = (31q - 159)/60\,480. \tag{2.6}$$

Then the local truncation error of the method $S(q)$ is

$$\mathrm{LTE}(x_n;q) = h^8 \Delta(x_n;q) + O(h^{10}) \tag{2.7}$$

with

$$\Delta(x;q) = C(q) y^{(8)}(x).$$

The schemes corresponding to $q = 1, 0$ and -1 were originally considered by Henrici [6, formulae (6.22), (6.76) and (6.77)] and further applied on certain occasions. $S(1)$, for instance, was applied in [3, 13] as a part of some special technique, while $S(-1)$ was at the basis of the investigations reported in [8, 11, 12]. We will also give some special attention to two particular schemes, namely $S(0)$ and $S(1)$. Indeed, Eq. (2.5) suggests that the weights of $S(q)$ result directly from the weights of $S(0)$ and $S(1)$ by the formal relation

$$S(q) = (1 - q)S(0) + qS(1), \tag{2.8}$$

with the meaning that

$$a_i = (1 - q)a_i^0 + qa_i^1, \qquad b_i = (1 - q)b_i^0 + qb_i^1 \tag{2.9}$$

with $i = 0, 1, 2, 3, 4$, where a_i^0, b_i^0 and a_i^1, b_i^1 are the weights of $S(0)$ and $S(1)$, respectively.

The numerical values are:

$$a_0^0 = a_4^0 = 1, \qquad a_1^0 = a_3^0 = -1, \qquad a_2^0 = 0,$$

$$b_0^0 = b_4^0 = 17/240, \qquad b_1^0 = b_3^0 = 29/30, \qquad b_2^0 = 37/40, \tag{2.10}$$

$$a_0^1 = a_4^1 = 1, \qquad a_1^1 = a_3^1 = 0, \qquad a_2^1 = -2,$$

$$b_0^1 = b_4^1 = 1/15, \qquad b_1^1 = b_3^1 = 16/15, \qquad b_2^1 = 26/15, \tag{2.11}$$

and it will be particularly useful to keep in mind that the middle term is absent in the left hand member of the algorithm (2.2) corresponding to $S(0)$ while the second and the fourth terms are absent for $S(1)$.

2.2. Stability condition

This condition imposes some restriction on the values to be taken for q. We recall here that scheme (2.2) is zero-stable if no root of the characteristic equation

$$a_0 d^4 + a_1 d^3 + a_2 d^2 + a_3 d + a_4 = 0$$

exceeds 1 in absolute value and the multiplicity of the roots of modulus 1 is at most two. For $S(q)$ the characteristic equation is

$$d^4 + (q-1)d^3 - 2qd^2 + (q-1)d + 1 = (d-1)^2(d^2 + (q+1)d + 1) = 0$$

and this shows that the conditions mentioned are satisfied only when $q \in (-3, 1]$. It can be also shown that when $q \in (-3, 1)$, $S(q)$ is *strongly* stable.

2.3. Periodicity interval

In essence, this parameter is useful in that it offers some indication on the magnitude of maximal h to be used in the integration of certain equations. The test equation is $y'' = -p^2 y$, for which $S(q)$ reads

$$(1 - b_0\lambda)(y_{n+2} + y_{n-2}) + (a_1 - b_1\lambda)(y_{n+1} + y_{n-1}) + (a_2 - b_2\lambda)y_n = 0,$$

with $\lambda = -p^2h^2$. Its characteristic equation is

$$(1 - b_0\lambda)(d^4 + 1) + (a_1 - b_1\lambda)(d^3 + d) + (a_2 - b_2\lambda)d^2 = 0. \tag{2.12}$$

Nonpositive values of λ are only considered and the periodicity interval $[P(q) < 0, 0]$ is defined as the range of λ's for which the roots of Eq. (2.12) obey the stability conditions. A knowledge of $P(q)$ allows us to predict that, for equations which are well simulated by the criterion equation, $S(q)$ will give reasonable results only if $h \leqslant (-P(q))^{1/2}/p$ and thus schemes for which $|P(q)|$ is the largest appear to be favored.

In Figs. 1 and 2 we give $P(q)$ and $\bar{C}(q) = C(q)/(b_0 + b_1 + b_2 + b_3 + b_4) = C(q)/(q+3)$, respectively, on the convergence interval. (This $\bar{C}(q)$ actually governs the behaviour of the accumulated error.)

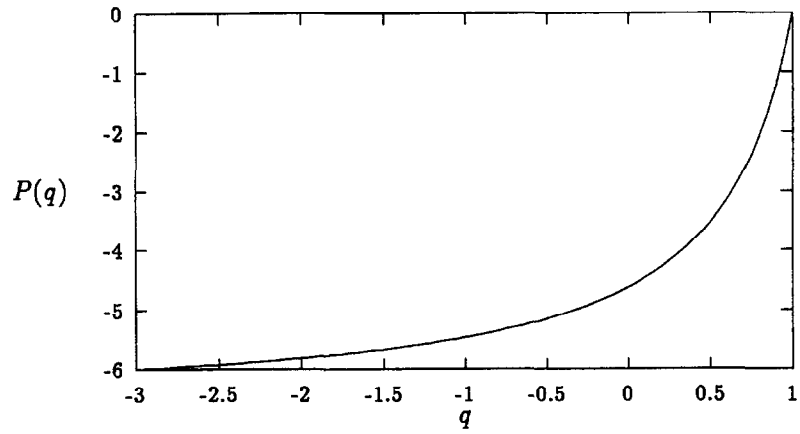

Fig. 1. Dependence on q of the lower limit P of the periodicity interval.

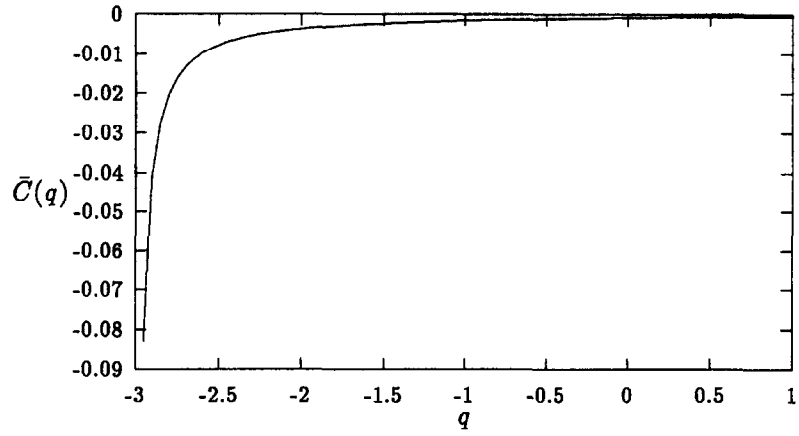

Fig. 2. Dependence on q of the accumulated error constant $\overline{C} = C(q)/(q+3)$.

We see that the schemes with q in the vicinity of 1 are preferable from the accuracy point of view but the methods with q toward -3 are the best with respect to the periodicity range criterion. However, if q is between -1 and 0.5, $P(q)$ and $\overline{C}(q)$ remain still close enough to their corresponding optimal values and for this reason in practical applications some priority should be given to such q.

3. Practical consequences

In this section we mention some ways in which the knowledge of the fact that the four step schemes form a family can be exploited to increase the computational efficiency.

3.1. Reduction of the number of iterations

Let us suppose that a value q^* was chosen in the convergence range of q and the corresponding $S(q^*)$ will be used to advance the solution in the four step interval centered at x_n i.e. to calculate y_{n+2} and f_{n+2} when all other quantities which enter Eq. (2.2) are known.

Since scheme $S(q^*)$ is implicit, the calculation should proceed iteratively, in a predictor–corrector mode $P(EC)^\mu E^{1-t}$, according to the notation used in [9]. Corrector C is, of course, $S(q^*)$ while a suitable predictor can be found inside the family of four step methods, as well. This means searching for a value Q such that $S(Q)$ is explicit, that is $b_0 = 0$. Using Eq. (2.5) we get $Q = -17$, i.e. $P = S(17)$ is the suited predictor.

The proposed predictor–corrector mode exhibits some particularities. To see them it should be first reminded that in general the main requirement on the two components of an efficient predictor–corrector procedure is that the order of the predictor is usually smaller but as close as possible to the order of the corrector; this secures a drastic reduction in the number of iterations. Seen from this perspective, the best pair is that in which the orders of the predictor and of the corrector are equal. If, in the latter case, it happens that the a-coefficients coincide in the predictor and the corrector, this mode enjoys the additional advantage of furnishing a measure of the LTE by Milne's device.

In all these considerations there is a tacit assumption: the predictor is a stable scheme. This assumption is clearly violated in our case: $Q = 17$ is outside the stability region. It follows that in our case the main role of the successive iterations consists in removing the instability of the predictor and therefore the type and the number of iterations must be settled in terms of this particularity.

The fact that the leading terms of the LTE in the predictor and the corrector are proportional suggests that the Newton–Raphson iteration is more convenient than the usual fixed point iteration. The Newton–Raphson iteration procedure also exhibits the advantage that for all $\mu \geqslant 2$ and any $t = 0, 1$ the stability properties of the $P(EC)^\mu E^{1-t}$ mode are identical to those of C alone, see [9]. This means that the instability of the predictor is entirely removed with two iterations only; any further iteration is redundant. For these reasons, only the versions PEC (i.e. with *one* function evaluation per step) and $P(EC)^2$ (i.e. with *two* function evaluations per step) are actually relevant in applications. The latter shares entirely the stability properties of the corrector $C = S(q^*)$ alone; these properties were investigated in Section 2. As for the stability properties of PEC, these are more modest; this version is zero-stable only for $-1 < q < 1$ and its periodicity area is only a small portion of the periodicity area of $S(q)$, Fig. 1, just below the segment $-1 < q < 1$. From the practical point of view this means that values $q^* > -1$ are mainly recommended since they are acceptable for both versions. The version of use is dictated by the requested accuracy. If low accuracies are required, the version with the largest stability area should be used and this is $P(EC)^2$. In contrast, if high accuracies are required, then PEC, i.e. with only *one* function evaluation per step, is sufficient. As a consequence, it can be said that the domain of high accuracies is the one where our method is expected to be specially efficient.

The proposed predictor $S(17)$ does not meet the necessary requirement for Milne's device. A convenient scheme can be, however, constructed without difficulty. We search for a six-step scheme (further on identified as M) with the desired structure

$$y_{n+2} + a_1(q^*)(y_{n+1} + y_{n-1}) + a_2(q^*)y_n + y_{n-2}$$
$$= h^2[\beta_1(q^*)y''_{n+1} + \beta_2(q^*)y''_n + \beta_3(q^*)y''_{n-1} + \beta_4(q^*)y''_{n-2} + \beta_5(q^*)y''_{n-3} + \beta_6(q^*)y''_{n-4}], \quad (3.1)$$

i.e. with the same a-coefficients as the corrector $S(q^*)$; a vanishing coefficient of y''_{n+2} is also imposed.

The technique to obtain the β coefficients is the same as for algorithm (2.2), i.e. a functional is associated to Eq. (3.1) and this is further required to identically vanish for any of the eight functions under Eq. (2.4). Note that Eq. (3.1) is automatically satisfied for $y=1$ and $y=x$; only the other six functions $y=x^n$, $n=2,3,\ldots,7$ actually generate equations for the six unknowns.

The resultant weights are

$$b_1(q^*)=\frac{167+9q^*}{120}, \qquad b_2(q^*)=\frac{11(-3+19q^*)}{240}, \qquad b_3(q^*)=\frac{143+q^*}{60},$$
$$b_4(q^*)=\frac{7(-17+q^*)}{120}, \qquad b_5(q^*)=\frac{17-q^*}{40}, \qquad b_6(q^*)=\frac{-17+q^*}{240}, \tag{3.2}$$

and the local truncation error of M reads

$$\mathrm{LTE}_M(x_n;q^*)=C_M(q^*)y^{(8)}(x_n)h^8+\mathrm{O}(h^{10}), \qquad C_M(q^*)=\frac{2(4125-221q^*)}{3.8!}. \tag{3.3}$$

As a matter of fact, scheme M can also be used in the predictor–corrector mode, in the place of $S(17)$. In our program, however, we opted for $P=S(17)$, to involve M only when estimating the error.

If y^M_{n+2} is the value furnished by M, Milne's device gives the simple approximation

$$\varepsilon(x_n;q^*)=\frac{C(q^*)}{C(q^*)-C_M(q^*)}(y^M_{n+2}-y_{n+2}) \tag{3.4}$$

for the local truncation error in $S(q^*)$.

3.2. Starting the integration

To start, the integration algorithm $S(q^*)$ requires four values as input, namely y_0, y_1, y_2 and y_3. The value y_0 is furnished by the initial condition and we assume that y_1 is also known. (In practice it can be generated by a one-step method of sufficient accuracy. DOPRIN subroutine can be used for this purpose.) The actual problem then consists of generating y_2 and y_3 with an accuracy of h^8. A succession of five steps is described below in which the Numerov and $S(1)$ schemes are conveniently used.

Step 1: Given y_0, y_1, f_0 and f_1, we calculate $y_{1/2}$ and $f_{1/2}$ by the Numerov method in the interpolation regime, i.e. with

$$y_0-2y_{1/2}+y_1=\tfrac{1}{8}h^2(f_0+f_1) \tag{3.5}$$

as predictor, and with

$$y_0-2y_{1/2}+y_1=\tfrac{1}{4}h^2\left(\tfrac{1}{12}f_0+\tfrac{5}{6}f_{1/2}+\tfrac{1}{12}f_1\right) \tag{3.6}$$

as corrector, in the $P(EC)^3$ mode, i.e. with three complete iterations. The resultant $y_{1/2}$ has an error proportional to h^6.

Step 2: Given $y_{1/2}, y_1, f_{1/2}$ and f_1 we calculate $y_{3/2}$ and $f_{3/2}$ by the Numerov method at $\frac{1}{2}h$ in the extrapolation regime, i.e. with the predictor $y_{3/2} - 2y_1 + y_{1/2} = \frac{1}{4}h^2 f_1$ and using three iterations. The leading term of the error in the resultant $y_{3/2}$ is also proportional to h^6.

Step 3: Given $y_0, y_{1/2}, y_1, y_{3/2}, f_0, f_{1/2}, f_1$ and $f_{3/2}$, we calculate y_2 and f_2 by $S(1)$ at $\frac{1}{2}h$ in the $P(EC)^2$ mode. Note that since the values of $y_{1/2}$ and $y_{3/2}$ do not appear explicitly in $S(1)$ (they intervene only as arguments in $f_{1/2}$ and $f_{3/2}$ and these are multiplied by $\frac{1}{4}h^2$) the leading term of the error in the resultant y_2 will be proportional to h^8, although $y_{1/2}$ and $y_{3/2}$ are less accurate.

Step 4: Given $y_{3/2}$, y_2, $f_{3/2}$ and f_2 we calculate $y_{5/2}$ and $f_{5/2}$ by the Numerov method with three iterations. The leading term of the error in the resultant $y_{5/2}$ is proportional to h^6.

Step 5: Given y_1, $y_{3/2}$, y_2, $y_{5/2}$, f_1, $f_{3/2}$, f_2 and $f_{5/2}$, y_3 and f_3 are evaluated by $S(1)$ at $\frac{1}{2}h$ also in the $P(EC)^2$ mode. The error in y_3 is of h^8, on the same basis as in Step 3.

The generation of the additional starting values is thus complete and it consists of 13 evaluations of f. As a matter of fact, this number of function evaluations is slightly smaller than the one to be required if y_2 and y_3 were computed by a one step method of comparable accuracy. DOPRIN program, for instance, needs eight function evaluations for each of the two values. We note for completeness that the above procedure represents only one possibility; for an alternative procedure see [3, Eqs. (2.6 a,b,c)].

3.3. Halving step sizes

Let us assume that the values of y_i, f_i, $i = n-1, n, n+1, n+2$ are known and that we want to go next with $S(q^*)$ at halved step size. The first pair to be calculated thus is $y_{n+5/2}$ and $f_{n+5/2}$ and its computation by $S(q^*)$ clearly requires eight input values; four of them are known, viz. $y_{n+1}, f_{n+1}, y_{n+2}$ and f_{n+2}, while the other four, viz. $y_{n+1/2}$, $f_{n+1/2}, y_{n+3/2}$ and $f_{n+3/2}$ must be calculated by a separate procedure. Such a procedure is described below.

We consider three versions $S(q_1)$, $S(q_2)$, $S(q_3)$ to be used at $\frac{1}{2}h$ in the four step intervals centered at x_n, $x_{n+1/2}$ and x_{n+1}, respectively. We first take the left-hand side of Eq. (2.2), write the involved y's as the head line and place the corresponding weights at the appropriate positions under each y. We also mention explicitly the argument of each weight and account for the symmetric weights.

The following array results:

	y_{n-1}	$y_{n-1/2}$	y_n	$y_{n+1/2}$	y_{n+1}	$y_{n+3/2}$	y_{n+2}
$S(q_1)$:	1	$a_1(q_1)$	$a_2(q_1)$	$a_1(q_1)$	1		
$S(q_2)$:		1	$a_1(q_2)$	$a_2(q_2)$	$a_1(q_2)$	1	
$S(q_3)$:			1	$a_1(q_3)$	$a_2(q_3)$	$a_1(q_3)$	1

(3.7)

We organize the right-hand side size the same way:

	$\frac{1}{4}h^2 f_{n-1}$	$\frac{1}{4}h^2 f_{n-1/2}$	$\frac{1}{4}h^2 f_n$	$\frac{1}{4}h^2 f_{n+1/2}$	$\frac{1}{4}h^2 f_{n+1}$	$\frac{1}{4}h^2 f_{n+3/2}$	$\frac{1}{4}h^2 f_{n+2}$
$S(q_1)$:	$b_0(q_1)$	$b_1(q_1)$	$b_2(q_1)$	$b_1(q_1)$	$b_0(q_1)$		
$S(q_2)$:		$b_0(q_2)$	$b_1(q_2)$	$b_2(q_2)$	$b_1(q_2)$	$b_0(q_2)$	
$S(q_3)$:			$b_0(q_3)$	$b_1(q_3)$	$b_2(q_3)$	$b_1(q_3)$	$b_0(q_3)$

(3.8)

In general, if we multiply the $S(q_i)$ row by a constant c_i, $i=1,2,3$ and then sum the elements in each column, we obtain a formula of the form $T=c_1S(q_1)+c_2S(q_2)+c_3S(q_3)$ which connects the values of y and of f at seven successive $\frac{1}{2}h$ spaced points. The formula is linear and homogeneous in the weights and so we can fix one constant from the very beginning, without affecting the generality. We thus take $c_1=1$ and address the problem of finding the values of the remaining five parameters c_2, c_3, q_1, q_2 and q_3 such that the resultant formula has some practical relevance.

Problem 1 consists in finding the parameters of two formulae of the T type to be used for evaluating $y_{n+1/2}$ and $f_{n+1/2}$ in a predictor–corrector process. The predictor should consist of an explicit formula in which $y_{n+1/2}$ is evaluated in terms of the mentioned known quantities. This is achieved by forcing the coefficients of $y_{n-1/2}$, $y_{n+3/2}$, $f_{n-1/2}$, $f_{n+3/2}$ and $f_{n+1/2}$ to vanish. We arrive at the system:

$$\begin{aligned} &a_1(q_1)+c_2=0, \qquad c_2+c_3a_1(q_3)=0, \qquad b_1(q_1)+c_2b_0(q_2)=0,\\ &c_2b_0(q_2)+c_3b_1(q_3)=0, \qquad b_1(q_1)+c_2b_2(q_2)+c_3b_1(q_3)=0. \end{aligned} \tag{3.9}$$

Its solution is given by the following formulae to be used in the order given below:

$$q_2=\frac{2b_0^0-b_2^0}{2(b_0^0-b_0^1)+b_2^1-b_2^0}, \qquad q_1=\frac{b_1^0-b_0(q_2)a_1^0}{b_1^1-b_0(q_2)a_1^0-b_1^0}, \tag{3.10}$$

$$c_2=-a_1^0(1-q_1), \qquad c_3=c_2, \qquad q_3=q_1.$$

With the values given in Eqs. (2.10) and (2.11) we get $q_1=q_3=-1531/37$, $q_2=-47/49$, $c_2=1568/37$ and $c_3=1$ and thus we have the final formula:

$$37(y_{n-1}+y_{n+2})+27(y_n+y_{n+1})-128y_{n+1/2}=\tfrac{1}{4}h^2[9(f_{n-1}+f_{n+2})+171(f_n+f_{n+1})], \tag{3.11}$$

which yields $y_{n+1/2}$ directly.

As for the corrector, this must be of a form in which the coefficients of $y_{n-1/2}, y_{n+3/2}, f_{n-1/2}$ and $f_{n+3/2}$ vanish, but there is no restriction on the coefficient of $f_{n+1/2}$. Therefore the first four equations in (3.9) remain unchanged while the fifth equation is fixed by a different criterion. We want the corrected values to be as accurate as possible and, since the local truncation error of T is of the form

$$\mathrm{LTE}_T=h^8(C(q_1)+c_2C(q_2)+c_3C(q_3))y_{n+1/2}^{(8)}+\mathrm{O}(h^9)$$

(this is because T is linear in the three components and each component has the local error of the form (2.7)) we require that the coefficient of h^8 vanishes, i.e.

$$C(q_1)+c_2C(q_2)+c_3C(q_3)=0. \tag{3.12}$$

Upon solving the system we get $c_2 = 3840/219$, $c_3 = 1$, $q_1 = q_3 = -3621/219$ and $q_2 = 38/5$ i.e. the following formula should be used for the corrector:

$$219(y_{n-1} + y_{n+2}) + 32805(y_n + y_{n+1}) - 66048y_{n+1/2} = \tfrac{1}{20}h^2[153(f_{n-1} + f_{n+2}) + 19683(f_n + f_{n+1}) + 134208f_{n+1/2}]. \tag{3.13}$$

Problem 2 consists in finding similar formulae for the calculation of $y_{n+3/2}$ and of $f_{n+3/2}$. The same procedure is applied with the results:

Predictor:

$$37y_{n-1} + 15y_n - 205y_{n+1} + 128y_{n+3/2} + 25y_{n+2} = \tfrac{1}{12}h^2(27f_{n-1} + 509f_n + 761f_{n+1} + 23f_{n+2}). \tag{3.14}$$

Corrector:

$$-773y_{n-1} - 1655y_n + 39165y_{n+1} - 67072y_{n+3/2} + 30335y_{n+2} = \tfrac{1}{12}h^2(-497f_{n-1} - 12311f_n - 20259f_{n+1} + 79424f_{n+3/2} + 6563f_{n+2}). \tag{3.15}$$

In our program the two pairs (3.11), (3.13) and (3.14), (3.15) were used in the $P(EC)^3$ mode.

4. Numerical illustration

All the procedures presented in this paper were aimed at making the application of the four-step methods as flexible as possible. A program has been written in which these procedures (i.e. generation of the additional starting values, halving/doubling stepsizes, LTE estimation and control) were included. The program, called MSM4, works in double precision arithmetic and its main features are as follows:

1. y_0 and y_0' are given as input. y_1 is calculated by DOPRIN at tolerance $1.d-14$. The other two starting values, y_3 and y_4, are generated as explained in Section 3. However, for the test cases which follow y_1 was given as the value of the exact solution.

2. The tolerance value TOL is also given as input. At each step the program computes the LTE estimate and this is compared to TOL. If $\mathrm{TOL}/256 \leqslant |\mathrm{LTE}| \leqslant \mathrm{TOL}$ the computation advances with the current h. The factor $1/256$ is in accordance with the $O(h^8)$ behaviour of the LTE. If $\mathrm{TOL} < |\mathrm{LTE}| \leqslant 2\mathrm{TOL}$ at six consecutive steps the current h is halved, while, if $|\mathrm{LTE}| > 2\mathrm{TOL}$ it is halved immediately. Finally, if $|\mathrm{LTE}| < \mathrm{TOL}/256$ at six consecutive steps, the stepsize is doubled.

3. The Jacobian of the system, needed to perform the Newton–Raphson iteration, is calculated and LU decomposed only once after every 20 consecutive steps.

We now report on two test cases which in our opinion have a rather high degree of difficulty. For all these runs the program used the PEC mode in the form $S(17)\ E\ S(q^*)$ with $q^* = 0$.

The two test cases are

$$y'' = -4x^2y - \frac{2z}{\sqrt{y^2+z^2}}, \qquad z'' = -4x^2z + \frac{2y}{\sqrt{y^2+z^2}}, \tag{4.1}$$

$x \geqslant x_0$, $x_0 = \sqrt{\pi/2}$, $y(x_0) = z'(x_0) = 0$, $y'(x_0) = -\sqrt{2\pi}$, $z(x_0) = 1$ with the exact solution $y(x) = \cos(x^2)$, $z(x) = \sin(x^2)$, see e.g. [2], and

$$y'' = -a^2 \sin^2(x) y + \frac{a \cos(x) z}{\sqrt{y^2+z^2}}, \qquad z'' = -a^2 \sin^2(x) z - \frac{a \cos(x) y}{\sqrt{y^2+z^2}}, \tag{4.2}$$

$$x \geqslant 0, \qquad y(0) = \cos(a), \qquad z(0) = \sin(a), \qquad y'(0) = z'(0) = 0$$

with the exact solution $y(x) = \cos(a \cos(x))$, $z(x) = \sin(a \cos(x))$.

Each of the two systems (for the second we used $a = 100$) were integrated up to $x_{\max} = 10$ for several values of TOL. The results were compared with the ones produced by the Runge–Kutta–Nyström based subroutine DOPRIN, see [5]. As a matter of fact, we are fully aware that subroutines based on higher order Runge–Kutta–Nyström versions do exist, e.g. subroutine D02LAF in [4]. However, we think that selecting DOPRIN for reference is just fair enough because this is the closest to our method with respect to the order.

The errors at $x_{\max} = 10$ and some additional relevant data are displayed on Tables 1 and 2. If these are analysed in terms of the total number of steps (tns) to reach one and the same accuracy, DOPRIN is clearly in advantage; its tns is approximately one quarter of that required by our program. However, this criterion has only a limited relevance because what it is usually understood by a single step is different for multistep and Runge–Kutta one-step methods. In the former only the values of y and f at the mesh points of the partition matter, while in the latter some intermediate points are also involved.

More realistic is the comparison in terms of the number of function evaluations (nfe); this is actually the typical criterion to compare various methods for efficiency. Tables 1 and 2 clearly show that MSM4 is in advantage, its nfe being approximately half of that for DOPRIN. The two programs were also compared at lower tolerances, i.e. TOL $= 10^{-3}$, 10^{-4}, 10^{-5}. For the last two values the nfe for our program remained about half of that for DOPRIN. For TOL $= 10^{-3}$, however, the PEC mode was no longer sufficient in MSM4 because the involved step sizes are big enough to make this mode unstable. MSM4 then activated the $P(EC)^2$ mode, which, as explained in Section 3, removes the instability of the predictor. The corresponding number of function evaluations

Table 1
Deviations $\Delta y(x) = |y_{\text{exact}}(x) - y_{\text{comput}}(x)|$, $\Delta z(x) = |z_{\text{exact}}(x) - z_{\text{comput}}(x)|$, at $x = 10$ for Eq. (4.1), total number of steps (tns), number of rejected steps (nrs) and number of function evaluations (nfe) for programs DOPRIN and MSM4 at several values of TOL

TOL	10(−6)	10(−6)	10(−7)	10(−7)	10(−8)	10(−8)	10(−9)	10(−9)
Method	DOPRIN	MSM4	DOPRIN	MSM4	DOPRIN	MSM4	DOPRIN	MSM4
$\Delta y(10)$	5.5(−7)	1.1(−6)	5.4(−8)	1.0(−7)	5.0(−9)	2.2(−8)	5.2(−10)	2.5(−9)
$\Delta z(10)$	8.8(−7)	6.5(−7)	6.6(−8)	6.6(−8)	5.2(−9)	1.9(−8)	4.6(−10)	2.0(−9)
tns	170	698	244	990	359	1191	482	1799
nrs	15	5	26	5	51	5	52	10
nfe	1361	762	1953	1068	2873	1269	3857	1915

Table 2
Same as in Table 1 for the test equation (4.2) with $a = 100$

TOL	10(−6)	10(−6)	10(−7)	10(−7)	10(−8)	10(−8)	10(−9)	10(−9)
Method	DOPRIN	MSM4	DOPRIN	MSM4	DOPRIN	MSM4	DOPRIN	MSM4
$\Delta y(10)$	8.1(−6)	2.7(−6)	7.0(−7)	5.7(−7)	5.8(−8)	2.5(−8)	4.7(−9)	9.6(−9)
$\Delta z(10)$	7.8(−6)	2.3(−6)	6.7(−7)	3.8(−7)	6.0(−8)	2.3(−8)	5.0(−9)	7.0(−9)
tns	1040	4579	1520	6306	2225	8699	2994	9874
nrs	94	18	176	43	314	25	319	20
nfe	8321	4741	12 162	6572	17 801	8907	23 953	10 066

per step is now equal to two and for this reason nfe required by MSM4 equals nfe required by DOPRIN.

We thus conclude that, in terms of the number of function evaluations, our program is as efficient as DOPRIN if low accuracies are required, to become twice more efficient at high accuracies. However, it is fair adding that such a conclusion may not be true if comparison is made in terms of the CPU time. For instance, when large systems are to be solved, the construction of the jacobian (this is needed in the iteration procedure of MSM4) is a time consuming operation and then DOPRIN is expected to be generally faster. Also, DOPRIN is the method to be chosen for situations which are demanding for methods with broader intervals of periodicity, for example, when only low accuracies are required. The reason is that the interval of periodicity for the best of the two methods used in DOPRIN (this is the method of order seven) is $[-46.6, 0]$, i.e. substantially larger than in Fig. 1.

5. Conclusions

Our main goal in this paper was to amend the current image on the multistep methods. In fact, while these methods are generally recognized as being simple, many users complain on their rigidity.

We have shown that, at least for the four step methods, such a drawback can be removed to a substantial extent; except that they are not self-starting (the value of $y(x_{\min}+h)$ should be furnished separately), they are sufficiently flexible with respect to the stepsize modification when the solution is propagated. We have also presented a procedure for the evaluation of the local error.

The numerical tests convincingly show that these ingredients are sufficient to generate a robust and competitive code to solve second order non-stiff systems of the form $y'' = f(x, y)$.

Acknowledgements

One of us (L. Gr. I.) wishes to thank Prof. J.P. Coleman for pertinent comments on this subject. He also thanks Prof. G. Vanden Berghe, Prof. H. De Meyer and Dr. M. Van Daele for stimulating discussions and for suggesting the second test case.

References

[1] A.C. Allison and X. Huang, An exponentially fitted two-step method with predictor–corrector form for the solution of nonlinear or coupled Schrödinger equations, *Comput. Phys. Commun.* **66** (1991) 211.

[2] D.G. Bettis, A Runge–Kutta–Nyström algorithm, *Celestial Mech.* **8** (1973) 229.

[3] J.R. Cash and A.D. Raptis, A high order method for the numerical integration of the one-dimensional Schrödinger equation, *Comput. Phys. Commun.* **33** (1984) 299.

[4] NAG Fortran Library, The Numerical Algorithms Group Limited, Mark 15 (1991).

[5] E. Hairer, S.P. Norsett and G. Wanner, *Solving Ordinary Differential Equations I, Non-stiff Problems* (Springer, Berlin, 1987).

[6] P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations* (Wiley, New York, 1962).

[7] L.Gr. Ixaru, *Numerical Methods for Differential Equations and Applications* (Reidel, Dordrecht, 1984).

[8] L.Gr. Ixaru and M. Rizea, Comparison of some four-step methods for the numerical solution of the Schrödinger equation, *Comput. Phys. Commun.* **38** (1985) 329.

[9] J.D. Lambert, *Numerical Methods for Ordinary Differential Equations* (Wiley, Chichester, 1991).

[10] J.D. Lambert and I.A. Watson, Symmetric multistep methods for periodic initial value problems, *J. Inst. Math. Appl.* **18** (1976) 189.

[11] A.D. Raptis, On the numerical solution of the Schrödinger equation, *Comput. Phys. Commun.* **24** (1981) 1.

[12] A.D. Raptis, Exponentially-fitted solutions of the eigenvalue Schrödinger equation, *Comput. Phys. Commun.* **28** (1983) 427.

[13] A.D. Raptis and J.R. Cash, A variable step method for the numerical integration of the one-dimensional Schrödinger equation, *Comput. Phys. Commun.* **36** (1985) 113.

[14] A.D. Raptis and T.E. Simos, A four-step phase-fitted method for the numerical integration of second-order initial value problems, *BIT* **31** (1991) 160.

[15] T.E. Simos, A four-step method for the numerical solution of the Schrödinger equation, *J. Comput. Appl. Math.* **30** (1990) 251.

[16] T.E. Simos, An explicit four-step phase-fitted method for the numerical integration of second order periodic initial-value problems, *J. Comput. Appl. Math.* **55** (1995) 125.

[17] T.E. Simos, A family of four-step predictor–corrector exponential fitted methods for the numerical solution of the Schrödinger equation, *J. Comput. Appl. Math.* **58** (1995) 337.

[18] T.E. Simos and A.D. Raptis, A fourth order Bessel fitting method for the numerical solution of the Schrödinger equation, *J. Comput. Appl. Math.* **43** (1992) 313.